CLUBE Z80

NUMERO UM -30/OUTUBRO/1982

COORDENADORES : maria irene E alberto fernandes
AV.BOAVISTA-832,2.T 4100 PORTO

INTRODUÇÃO

Este é o segundo número do boletim do Clube Z-80. É cedo ainda para fazer o balanço desta iniciativa, pois ela depende fundamentalmente dos seus aderentes, e da sua capacidade para criar "coisas novas".

Neste momento ninguém fez nenhuma crítica negativa, quer ao aparecimento do Clube, quer em relação ao boletim.

Mas as críticas, as correcções, as ideias novas terão de surgir, sob pena de que a publicação ou a actividade do Clube sejam coisas dirigidas automaticamente sem receber o sinal dos receptores, significativo dos resultados obtidos.

Os aderentes cobrem já os distritos do Porto, Viana do Castelo, Braga,Coimbra, Castelo Branco, Setúbal,Aveiro, Lisboa, Faro e Funchal, mas o número de assinantes do boletim em relação às despesas que a sua publicação (modestíssima) acarreta é ainda insuficiente.

Vai ser necessário um bom esforço para a divulgação do Clube e aumento urgente do número de aderentes a esta iniciativa.

No próximo número será iniciada a divulgação dos princípios de Programação Estruturada - Novas linguagens a usar com o ZX81 - As "ferramentas do Programador" (novos programas) - As ligações externas do ZX81 (hardware) e programas diversos.

Pensamos que no próximo número será dado um salto qualitativo no conteúdo do boletim. Esperemos é que os utilizadores do ZX81 também assim o entendam.

U T I L I Z A Ç Ã O D A M E M Ó R I A

Tem havido no mercado da informática uma grande procura das memórias ZX-81. Neste artigo, tratamos de questões relacionadas com o modo de utilizar ao máximo a memória, em função do tipo de trabalho que executar. Serão levantadas algumas questões sobre memórias RAM, a que responderemos.

• Tenho uma memória de 1K RAM e, frequentemente, a capacidade de memória de que disponho é insuficiente para trabalhar determinados programas. Existe algum processo para determinar o espaço necessário a um programa? É possível encurtar um programa sem alterar a sua função?

A resposta à primeira questão é não. O espaço necessário ao programa é dado assim:
PRINT PEEK 16396+256*PEEK 16397-16509.
Esta linha determina o ponto em que acaba a área do programa - contido em D FILE em 16396 e 16397 - e subtrai o ponto de início da área do programa - isto é, 16509; Mas, se se quer executar o programa e exibir a informação, então será necessário mais espaço.
No ZX81, o ficheiro ocupa 25 bytes e mais um por cada caracter no écran - - incluindo espaços em branco originados por PRINT AT, TAB e vírgulas em instruções com PRINT. É necessário dispôr também de uma certa capacidade para incluir a área das variáveis, a zona de cálculos e o "stack" (pilha) da máquina e dos GOSUB.
A quantidade de espaço livre determina-se, em qualquer altura, achando a diferença entre o ponto de armazenagem (SP) e o ponto de STKEND - em 16412. Infelizmente, o ponto SP pode ser obtido usando apenas uma rotina em linguagem máquina.
De qualquer modo, a utilização da memória não é uniforme durante a execução e, portanto, a quantidade de espaços livres varia também; Assim, o nosso conselho é que se quiser saber o espaço que o seu programa ocupa, deverá observar D FILE, dando entrada a PRINT PEEK 16396 +256* PRINT PEEK 16397.
Quanto maior for o seu programa, melhores serão os resultados. Em alternativa, se quiser ter uma idéia da área ocupada após a execução de um programa, observe STKEND dando entrada a PRINT PEEK 16412+256* PRI PEEK 16413.
A diferença entre o resultado e RAMTOP - - 17408 na memória de 1K ZX-81 e 32768 na 16K RAM - dá a quantidade aproximada de espaços livres, incluindo cerca de 100 bytes para armazenagem.
Há muitas tácticas para tornar um programa mais curto sem alterar a sua função, mas para que os resultados sejam satisfatórios é necessário compreender a linguagem usada para armazenar programas em BASIC. Há três caminhos:
- Cada caracter - letras, sinais de pontuação, caracteres gráficos - ocupa 1 byte, tal como PRINT, LET, FOR.
- Os números são inseridos quer em forma caracteres, quer em forma numérica. No primeiro caso, usa-se um byte por cada dígit seguido de um byte com código 126, e depo 5 bytes contendo a forma numérica.
- Cada linha tem 5 bytes: 2 que contêm o número da linha, seguidos de outros 2 que incluem a extensão do resto da linha e, finalmente, 1 byte que contém o número decimal 118 no fim da linha (equivalente ao New Line).
Para ter uma idéia de como concretizar estas normas, ponha o ZX81 a funcionar e introduza:

```
10 FOR A = 16509 TO 16548
20 PRINT CHR PEEK A
30 NEXT A
```

Este programa exibe o conteúdo dos primeiros 40 bytes da sua área.
Para tornar um programa mais curto recomendamos:
- Evite instruções REM e faça as instruções PRINT o mais pequenas possível
- Se possível, use funções para estabelecer valores de variáveis; Por exemplo: LET A = CODE"K" é preferível a LET A = 48

- Reduza ao mínimo o número de variáveis, usando-as para outros objectivos. Os contadores em circuito fechado usam uma grande quantidade de espaço na área das variáveis.
- Use letras apenas para nomes de variáveis.

. Qual é a fórmula para o tamanho de um ARRAY?

A fórmula é: 4+2* nº de dimensões + 5* nº total de elementos.
Um ARRAY B (2, 5, 6) tem 3 dimensões e 2* 5* 6 = 60 elementos. Daí que 4+2*3+5* 60 = 310 bytes.

As fórmulas equivalentes para as outras variáveis são:
NÚMERO: 5+ um byte por cada caracter do número
VARIÁVEL DE CONTROLE DO CICLO: 18 bytes
STRING: 3 + um byte por cada item do string.
ARRAYS DE CARACTERES: 4+2* nº de dimensões + número de elementos.

. Um programa realisado com ZX81 com ou sem 16K RAM pode ser usado com qualquer tipo de memória RAM?

Sim, desde que se tenha capacidade suficiente de memória para armazenar o programa. Mas não há quaisquer problemas em passar um programa com 16K se só se possuir por exemplo, 4K RAM.

. Os programas para memórias RAM superiores a 16K são especiais?

Não. Pode ir até 48K de memória, começando no ponto 16384 sem precisar de "software", embora isso apresente dois problemas. O primeiro é que o ZX81 precisa, pelo menos, de 16K de memória e se se tiver mais é necessário re-estabelecer o ponto RAMTOP quando ligar a alimentação.

Para isso, entre com:
POKE 16389,4* M+64
-256* INT((4* M+64)/256)
seguido de NEW
em que M é a quantidade de memória disponível. P. ex. se acrescentar 32K de memória, faça POKE 16389,192
NEW
O outro problema é que o ficheiro exibido tem que ficar retido na memória 16K. Portanto, se não usar "software" especial, fica reduzido a programas BASIC mais pequenos que 15K. O resto da memória pode ser aproveitado para ARRAYS. Na linha seguinte damos uma idéia do tamanho do seu programa:
PRINT PEEK 16397/4 - 16;"K".

. O que acontece se se adicionar RAM entre 8K e 16K?

O Sinclair ROM usa pontos de 0 a 8191; 8192 a 16383 não são geralmente usados. A memória RAM de 1K e 16K e outras que sejam adicionadas, usam normalmente 16384 ou mais. Actualmente, algumas das memórias produzidas em Inglaterra estabelecem memória RAM entre 8192 e 16383.
A vantagem desta simplicidade é que a área não é acessível ao SINCLAIR BASIC, a menos que se usem os comandos PEEK e POKE. Portanto, pode ser usada para armazenar dados sem correr o risco de serem "sobre-escritos". Aliás, eles não desaparecerão desde que o programa seja passado de uma cassete e a área pode ser usada para passar dados entre programas.

. O que significa especificamente "paginar"?

O microprocessador de apoio do ZX81 pode basear-se apenas em posições de 64K. Paginar é um modo simples de separar RAM, geralmente - mas não necessariamente - em blocos ou "páginas" de 64K, para que os processadores possam passar de uma "página" a outra.

INICIAÇÃO À LINGUAGEM MÁQUINA

Parece ter-se criado um mito à volta da linguagem máquina do ZX81. É comum pensar-se – e não compreendemos a razão – que a programação nessa linguagem é difícil.
Vamos tentar desfazer essa idéia?

Para começar, só precisa de um pequeno programa (BASIC), extraído do manual "Mastering Machine Code on Your ZX81". Eis um:

```
10 INPUT X
20 LET A$ = " "
30 IF A$ = " " THEN INPUT A$
40 IF A$ = "S" THEN STOP
50 POKE X, 16*CODE A$ +
   CODE A$(2) - 476
60 LET X = X + 1
70 LET A$ = A$(3 TO)
80 GOTO 30
```

É claro que não iremos aqui ensinar toda a linguagem máquina de um momento para o outro – daria "pano para mangas". Para já, vamos apenas fornecer-lhe uma ou duas rotinas que podem ser usadas com os seus programas BASIC. Experimente esta:
Dê entrada (Load) do programa acima referido e, a seguir, introduza esta linha

```
1 REM 1234567890234567 89
```

Agora faça correr (Run) o programa e introduza:

```
16514            Endereço do có-
                 digo Máquina
"2A0C40"         LD HL,
                 (D,FILE)
"0618"           LD B,
                 vinte e quatro
"23"  CICLO:     INC HL
"7E"             LD A,(HL)
"EE80"           XOR 80h
"FEF6"           CPF6h
"2803"           JR Z,SAÍDA
"77"             LD (HL),A
"18F5"           JR CICLO
"10F3" SAÍDA:    DJNZ CICLO
"C9"             RET
```

Não precisa compreender o que está escrito na coluna da direita. Basta introduzir o que é apresentado na esquerda. Depois, páre o programa ("S").
A seguir acrescente estas linhas:

```
100 IF INKEY$<>" "THEN GOTO 100
105 IF INKEY$ = " "THEN GOTO 105
110 RAND USR 16514
120 GOTO 100
```

Faça RUN 90 e repare no que acontece quando accionar qualquer tecla várias vezes seguidas. As linhas 10 a 80 podem, realmente ser eliminadas, sendo suficiente manter a linha 1. A partir deste momento, em qualquer altura que a máquina depare com a instrução RAND USR 16514, a "magia" da linguagem máquina entra em acção.
Experimente agora o seguinte: Elimine todas as linhas, excepto a primeira. Depois acrescente:

```
10 FOR I = 1 TO 100
20 PRINT "três espaços seguidos por três
   espaços inversos";
30 NEXT I
40 FOR I = 1 TO 50
50 NEXT I
60 RAND USR 16514
70 GOTO 40
```

Passe este programa e armazene-o.
Agora, se quiser avançar um bocadinho mais aqui está uma rotina que fará mover um ponto dentro do écran, saltando dentro dos seus limites.
Pode preparar um tipo de jogo de evasão recorrendo a esta rotina.
Introduza o programa de carga em linguagem máquina que apresentámos no início do artigo e adicione esta linha:

```
1 REM 12345678901234567890123456789012345
      6789012345678901234567890123456789012
      345678901234567890123
```

Agora faça correr o programa e introduza (contando "/" como "newline"):

```
16516/0101/2A8240/3600/3A8440/3D/2002/23/
2B/7E/FE80/200B/2A8240/4A8440/ED44/328440
/228240/3A8540/3D/2006/11DFFF/19/1804/
112100/19/7E/FE80/200B/2A8240/3A8540/ED44/
328540/010000/7E/FE08/2009/03/3A8540/ED44/
328540/228240/3634/C9/S
FE80/20CB/2A8240/3A8440
```

Agora pode eliminar todas as linhas depois da 10.
Para compreender como tudo isto é executa

→

do, acrescente:

```
10 PRINT "trinta e dois espaços inver-
   sos"
20 PRINT "espaço inverso+30-espaços+
   espaço inverso"
30 como 20
40 PRINT "espaço inverso+30 A gráfi-
   cos+espaço inverso"
50 como 40
60 FOR I = 1 TO 10
70 PRINT "espaço inverso+30 espaços+
   espaço inverso
75 NEXT I
76 PRINT "32 espaços inversos"
80 LET X = PEEK 16396 + 256*
   PEEK 16397 + 200
90 POKE 16514,X - 256*INT (X/256)
100 POKE 16515,INT (X/256)
110 LET A = USR 16518
120 GOTO 110
```

As linhas 10 a 76 apenas preenchem o écran; pode substituí-las por qualquer outra rotina que prefira

As linhas 80 a 100 são rigorosamente necessárias ao funcionamento da linguagem máquina ("+200" pode ser mais ou menos "+ qualquer coisa" - determina a posição inicial do ponto) e a parte fulcral está em 110 e 120. A instrução LET A = USR 16518 só faz mover o ponto ao longo de um quadrado. Se ordenar PRINT A, verifica que A é sempre zero, a menos que o ponto atinja o limite do écran, caso em que será igual a 1.

Como verifica, o ciclo 110 a 120 é o "jogo". Cabe-lhe a si a tarefa de o tornar mais aliciante; desde que não faça alterações à linguagem máquina e utilize BASIC, atendendo a que LET A = USR 16518 significa "mover o ponto num quadrado" - não terá problemas!

No próximo número continuaremos com o estudo da linguagem máquina, aplicada ao microprocessador Z80.

SECÇÃO DO LEITOR

OPINIÕES... DÚVIDAS...SUGESTÕES...COMENTÁRIOS...OPINIÕES...DÚVIDAS...SUGESTÕES...COMENTÁRIOS...

CRIÁMOS ESTA SECÇÃO PARA SI. COLABORE.

ESCREVA-NOS!

Serafim Miguel Guimarães, de 16 anos, escreveu-nos sugerindo uma versão diferente para o programa LASER que apresentámos no número anterior.
"Achei imensa piada ao jogo e resolvi fazer-lhe umas alterações. Depois destas, o programa ficou assim:"

```
1 REM"LASER"
2 PRINT AT 9,0;"--------------------------------"
3 PRINT AT 10,13;"LASER"
4 PRINT AT 11,0;"--------------------------------"
5 PRINT AT 19,26;"LOG"
6 PAUSE 200
7 CLS
8 LET G=0
9 LET P=0
10 LET A=0
20 LET J=20
30 LET K=10
50 LET X=INT (RND*18)+2
60 LET A=A+1
70 IF A=21 THEN GOTO 260
80 LET Y=30
90 PRINT AT K,0;CHR$ 130;CHR$ 128;AT X,Y;"X"
100 IF J<0 THEN GOTO 150
110 IF INKEY$="7" THEN LET K=K-1
120 IF INKEY$="6" THEN LET K=K+1
130 IF INKEY$="8" THEN PRINT AT K,2;"**************"
140 IF INKEY$="8" THEN LET J=J-1
150 LET Y=Y-1.5
152 IF J<=5 THEN PRINT "FUEL";J
155 IF J=0 THEN GOTO 240
160 IF Y=3 THEN LET G=G+1
170 IF G=8 THEN GOTO 240
180 IF Y=3 THEN GOTO 50
190 IF INKEY$="8" AND K=X AND Y<21 THEN GOTO 220
200 CLS
210 GOTO 90
220 PRINT AT X,Y;CHR$ 189
225 LET P=P+1
230 GOTO 50
240 PRINT "FIM DO JOGO"
241 PRINT "PONTOS= ";P
242 PRINT "FUEL= ";J
243 IF G>=8 THEN STOP
244 IF P>=5 THEN GOTO 260
245 STOP
260 PRINT "BONUS"
270 PAUSE 150
275 CLS
276 LET G=5
280 GOTO 10
999 SAVE "LASER"
```

"O jogo assim acaba quando:
1 - Acabam os 20 disparos possíveis
2 - Há 8 "X" que se aproximam demasiado da nossa base
Em caso de se fazer 5 pontos, há um bónus. Neste caso, há mais 20 disparos para dar e só se pode "deixar fugir" 3 "X". Haverá, depois, sempre bónus até que acabem os disparos (FUEL), ou o valor G atinja 8."

PROGRAMA PARA CALCULAR VALORES DE VENDAS A PRESTAÇÕES E PREÇOS DE VENDA A PÚBLICO

Este programa que nos foi fornecido por um membro do Clube - o Sr. Fernando Marques - é uma prova da validade e interesse desta associação e da forma como pode evoluir, no sentido de auxiliar os utilizadores do ZX81 a tirarem melhor rendimento do uso da sua máquina.

```
REM =VENDAS A PRESTACOES=              =PRECO V/PUBLICO=
  1 REM ELABORADO POR FERNANDO MARQUES**GONDOMAR/OUTUBRO-1982
  2 REM
  5 CLS
  6 REM "F.M."
  7 REM ESTE PROGRAMA CALCULA I.T. C/15 POR CENTO
 10 PRINT "ESTE PROGRAMA CALCULA:"
 11 PRINT "----------------------"
 12 PRINT AT 9,2;"1 - VENDAS A PRESTACOES"
 13 PRINT AT 11,2;"2 - PRECO DE VENDA AO PUBLICO"
 15 PRINT AT 17,2;"ESCOLHA O NUMERO PRETENDIDO"
 17 INPUT X$
 19 IF X$="1"THEN GOTO 99
 21 IF X$="2"THEN GOTO 4490
 99 CLS
100 LET M$="VENDAS A PRESTACOES"
110 LET N$="-------------------"
120 LET O$="VALOR DA VENDA"
130 LET P$="PRAZO/MESES"
140 LET Q$="ENTRADA INICIAL(0/0)"
150 LET R$="ENTRADA INICIAL(FIXA)"
160 LET S$="CERTO?  (S/N)"
170 PRINT TAB 7;M$
180 PRINT AT 1,7;N$
190 PRINT AT 6,0;"INDIQUE:"
200 PRINT AT 10,0;O$
210 PRINT AT 12,0;P$
220 PRINT AT 14,0;Q$
230 PRINT AT 16,0;R$
240 LET B$="a"
260 LET D$="                                "
280 LET M=10
290 PRINT AT 19,2;D$
300 INPUT A
301 PRINT AT 19,2;"=";A;"$0=";TAB 13;S$
302 INPUT X$
303 IF X$<>"S"THEN GOTO 290
310 PRINT AT M,28;B$
315 LET M=M+2
335 PRINT AT 19,2;D$
340 INPUT T
343 PRINT AT 19,2;"=";T;"-M=";TAB 9;S$
345 INPUT X$
347 IF X$<>"S"THEN GOTO 335
350 PRINT AT M,28;B$
375 PRINT AT 19,2;D$
380 INPUT H
381 REM
```

```
382 REM
383 PRINT AT 19,2;"=";H;"P/C=";TAB 9,S$
385 INPUT X$
401 IF X$<>"S"THEN GOTO 375
420 IF H>=25THEN GOTO 471
421 LET M=M+2
422 PRINT AT M,28;B$
423 PRINT AT 19,2;D$
424 INPUT J
432 IF J<=(0.25*A)THEN GOTO 424
434 PRINT AT 19,2;"=";J;"$0=";TAB 13;S$
435 INPUT X$
437 IF X$<>"S"THEN GOTO 423
440 LET M=M+2
450 PRINT AT M,28;B$
475 CLS
500 LET M=2
505 PRINT TAB 7;M$
510 PRINT AT 1,7;N$
515 PRINT AT 2,0;O$
520 PRINT AT 4,0;P$
525 PRINT AT 6,0;Q$(1TO 15)
530 PRINT AT 8,0;"LIQUIDO"
535 PRINT AT 10,0;"ENCARGOS"
540 PRINT AT 12,0;"LIQUIDO+ENCARGOS"
545 PRINT AT 14,0;"PRESTACAO MENSAL"
550 PRINT AT 16,0;"SELO/CONTRATO"
555 PRINT AT 18,0;"LETRA+SELO"
700 LET M=2
705 LET C=A
706 LET C$="$0"
710 GOSUB 3000
715 LET C=T
716 LET C$="-M"
720 GOSUB 3000
725 LET B=VAL "INT (H*A/100)"
735 IF H=0THEN LET B=J
740 LET C=B
743 LET C$="$0"
745 GOSUB 3000
750 LET I=A-B
755 LET C=I
760 GOSUB 3000
765 LET K=VAL "INT (0.285*I*T/12)"
770 LET L=I+K
775 LET N=VAL "(INT (((L/T)/10)+5))*10"
780 LET O=VAL "N*T-I"
790 LET C=O
800 GOSUB 3000
805 LET P=VAL "O+I"
810 LET C=P
815 GOSUB 3000
820 LET C=N
822 REM
```

```
925 GOSUB 3000
930 LET W=VAL "120+(INT ((4*A/1000)+0.5))"
935 LET C=W
940 GOSUB 3000
945 LET Y=VAL "INT ((3*N/1000)+0.5)"
950 LET C=Y
953 GOSUB 3000
900 PRINT AT 21,0;"=SE JA TOMOU NOTA,PRIMA (N/L)="
955 INPUT X$
960 CLS
970 RUN
3000 LET V$=STR$ C
3010 LET U=LEN V$
3020 PRINT AT M,28-U;C;C$
3025 PRINT AT 19,2;D$
3040 LET M=M+2
3050 RETURN
4490 CLS
4500 PRINT TAB 5;"CALCULO DO PRECO V/P"
4505 PRINT AT 1,5;"---------------------"
4510 PRINT AT 3,5;"OPCOES:"
4515 PRINT AT 7,0;"1 - P/CUSTO C/ IT INCLUIDO - 25"
4520 PRINT AT 9,0;"2 - P/CUSTO C/ IT INCLUIDO - 30"
4525 PRINT AT 11,0;"3 - P/CUSTO S/ IT INCLUIDO - 25"
4530 PRINT AT 13,0;"4 - P/CUSTO S/ IT INCLUIDO - 30"
4535 PRINT  AT 17,0;"ESCOLHA A OPCAO;"
4540 PRINT AT 19,0;"QUANDO TERMINAR OS CALCULOS:     =PRIMA (0) E DEPOIS <N/L>="
4542 LET M$="\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\"
4545 INPUT  X$
4550 IF X$="1"THEN GOTO 4570
4555 IF X$="2"THEN GOTO 4635
4560 IF X$="3"THEN GOTO 4685
4565 IF X$="4"THEN GOTO 4740
4570 CLS
4580 PRINT "IT INCLUIDO -25"
4585 PRINT M$
4590 INPUT A$
4595 IF  A$="0"THEN GOTO 5
4600 LET C=VAL A$
4605 LET B=VAL "(INT ((1.25*C)+0.6)/10)*10"
4610 LET U$=STR$ B
4615 LET U=LEN U$
4620 LET V=LEN A$
4625 PRINT TAB 10-V;A$;TAB 25-U;B;".0"
4630 GOTO 4590
4635 CLS
4640 PRINT "IT INCLUIDO -30"
4645 PRINT M$
4650 INPUT A$
4653 IF A$="0"THEN GOTO 5
4655 LET C=VAL A$
4660 LET B=VAL "(INT ((1.30*C)+0.6)/10)*10"
4665 LET U$=STR$ B
4670 LET U=LEN U$
4675 LET V=LEN A$
4680 PRINT TAB 10-V;A$;TAB 25-U;B;".0"
4683 GOTO 4650
4685 CLS
4686 REM
```


(Continua na pág. 12)

PROGRAMA PARA DETERMINAR A RAÍZ DE UMA FUNÇÃO PELO MÉTODO NEWTON

EXEMPLO CALCULADO : $f(x) = x^3 - 2x^2 - x + 2$

Supomos que $x_o = 3$ e $x = 2$

$h = 10^{-4}$

```
1 REM
10 REM "PROGRAMA PARA DETERMINAR A RAIZ DE UMA FUNCAO PELO METODO DE NEWTON"
20 PRINT"INTRODUZA O VALOR INICIAL X0"
30 INPUT C
35 PRINT"X0=";C
40 PRINT"INTRODUZA O INCREMENTO H"
50 INPUT A
60 PRINT"H=";A
70 LET X=C
80 GOSUB 500
90 LET Y=B
100 LET X=A+C
110 GOSUB 500
120 LET D=C
130 LET C=D-A*Y/(B-Y)
140 IF ABS(D-C)>=1E-8 THEN GOTO 70
150 PRINT"A RAIZ DA EQUACAO E X=";C
160 STOP
500 LET B=((X-2)*X-1)*X+2
510 RETURN
```

Neste programa, faz-se a determinação da raiz de uma função pelo metodo de Newton

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Quando o valor absoluto da diferença entre x_n e x_{n+1} se torna inferior a 10^{-8}, x_n é considerada como raiz e o cálculo fica terminado.

A definição da função é sempre feita na linha 500.
Para outra função, terá de ser introduzida na linha 500, antes de correr o programa.

Programa preparado pela Engª Teresa M. Ferreira

T A B E L A   D E   C A R A C T E R E S

Sendo o ZX81 um computador digital, não podemos admirar-nos pelo facto de ele realizar tudo com números. Todos os caracteres são formados por uma grelha de pontos oito-por-oito. Um espaço do caracter tem todos os pontos eliminados (branco), e um espaço inverso tem-nos exibidos (preto).
Começando na posição 7680 em ROM, cada caracter possui 8 bytes consecutivos que determinam a sua natureza. Cada byte representa uma série única de caracteres; se convertermos esse byte de decimal 0 a 255, em binário, 0000 0000 a 1111 1111, é revelado o conjunto de pontos que são excluídos ou que permanecem.
Estudámos o programa "Character Table Printer", no sentido de o tornar mais simples. Ele mostra a localização em ROM, o seu conteúdo decimal, o conteúdo convertido em binário e, finalmente, o binário convertido em espaços e espaços inversos.

```
1 REM TABELA DE CARACTERES <printer ZX >
10 FOR A=7680  TO 8192 STEP 8
15 FOR C=0 TO 7
20 LET D=PEEK (A+C)
30 PRINT A+C;"   ";D
40 FOR B=31 TO 24 STEP -1
50 PRINT AT C,B-12;D-2*INT (D/2)
60 PRINT AT C,B;CHR$((D-2*INT(D/2))*128)
70 LET D=INT(D/2)
80 NEXT B
90 NEXT C
100 PAUSE 100
110 POKE 16437,255
120 CLS
130 NEXT A
```

Se quiséssemos alterar o caracter estabelecido, teríamos que considerar a decisão com 256(32x8) pontos sobre 192(24x8) pontos inferiores. Não se pode ordenar POKE em ROM, mas há outras maneiras. A Sinclair estabelece um modo que permite imprimir informações gráficas.
A função L PRINT é executada, lendo um caracter da unidade de impressão - das posições 16444 a 16476, encontrando o conjunto de pontos do quadro em ROM, e enviando-os depois à impressora.
Jogar com o Sinclair implica movimentar RAMTOP para deixar um espaço de 256 bytes, depois copiar a rotina L PRINT de ROM para RAM, que não é a zona superior a RAMTOP. A seguir, esta rotina é alterada cuidadosamente, de modo que os seus caracteres em vez de começarem na posição 7680, começam a partir de 32255, a área superior a RAMTOP.
Nesses 256 bytes há capacidade para 32 caracteres, que tem a dimensão da unidade impressora. Podemos incluir os pontos que desejarmos nos bytes acima de RAMTOP, preencher a unidade de impressão com caracteres de 0 a 31, chamar a nossa rotina especial L PRINT e, imediatamente, eles estarão na impressora. É necessário um "hardware" especial, se quiser exibir no seu televisor os efeitos destas resoluções.

CÁLCULO NUMÉRICO UTILIZANDO LOGARITMOS

As funções trigonométricas SIN e COS são geralmente necessárias em casos que envolvam círculos, o que as torna muito importantes, mesmo que você dê pouca atenção à trigonometria.

Vamos aqui ocupar-nos das funções EXP e LN. Contrariamente às funções trigonométricas, estas só são necessárias em combinações de números e cálculos com fórmulas matemáticas.
A instrução EXP X corresponde à fórmula matemática exp(x) ou, mais vulgarmente, e^x. LN Y é o logaritmo natural, ln(y) , ou $\log_e y$.
Potencias e logaritmos são muito usados em cálculos científicos. Aparecem em estatísticas, descargas radioactivas, fórmulas químicas, e em tantos outros campos que, provavelmente, virá um dia a trabalhar com elas.
Para facilitar, usaremos a anotação A**B que significa "A com expoente B". Se B for um número inteiro positivo, então A**1=A; A**2=A*A; A**3=A*A*A , etc. Se B não for número inteiro, o expoente determina-se pela equação A**(B+C) = = (A**B)*(A**C). Por exemplo, se X = = 10**(0,5+0,5) = 10**1 = 10 , tal que X será a raiz quadrada de (10).
A função EXP calcula as potencias de um certo número, a que os matemáticos chamam "e", aproximadamente igual a 2,7183. Se ordenar PRINT EXP(1), este número será apresentado ainda mais pormenorizado. Geralmente, EXP B é a potencia Bª de "e" ou seja, EXP(B) = e**B.
Esta numeração começou a tornar-se importante a partir de meados de 1594,com os trabalhos logarítmicos de John Napier Os símbolos foram introduzidos por Leonard Euler em 1728.
Uma maneira de compreender como realmente eles funcionam é fazer cálculos com juros compostos.
Por exemplo, se 1$00 for investido à taxa de juroanual de 100%, composto anualmente, duplica em cada ano. Em N anos , terá aumentado até $ 2**N.
Suponha agora que o juro passa a ser aplicado não ao ano, mas em cada décima parte do ano - então a taxa deverá ser de 10%. Ao cabo de 0,1 anos, o investimento subiu para $1,1; ao fim de 0,2, somam-se mais 10%, obtendo 1,21$; depois 1,331$, 1,4641$ e assim sucessivamente.
Ao fim de 1 ano, equivale a $(1,1)**10, cerca de 2,594, e ao fim de N anos será $ (1,1)**(10*N).
Juros de 1% ao ano, compostos em cada 0,1 do ano, originam $(1,01)**100,ou 2,7048 ao fim de 1 ano e $(1,01)** (100*N) ao fim de N anos.
Continuando com este processo, compondo juros cada vez mais pequenos a intervalos cada vez maiores, ao fim de 1 ano o total aproxima-se cada vez mais de 2,718. E ao fim de N anos será cerca de$ e**N=EXP N
Quanto mais curtos forem os intervalos,maior será a aproximação, embora os limites aritméticos do ZX81 tornem mais difícil verificar isso directamente. Se experimentar (1,000001)**1000000 obterá 2,7176683 que é uma aproximação errada, devida a erros nos cálculos de arredondamento.
Há uma fórmula matemática para "e" que explica que (1+1/N)**N se aproxima muito de "e" quando N é grande.
Experimente o seguinte:

```
1 FOR I = 1 to 5
20 LET N = 10**I
30 LET E = (1+1/N)**N
40 PRINT N,E
50 NEXT I.
```

A função LN é o inverso de EXP, isto é, se EXP X=Y então Y=LN X. Este programa demonstrará isso:

```
10 FOR X = 1 to 20
20 PRINT X, EXP LN X
30 NEXT X.
```

Segundo os matemáticos, as propriedades mais importantes destas duas funções são provavelmente o facto de elas satisfazerem as equações:
EXP(A+B) = (EXP A)*(EXP B)
LN(A*B) = (LN A) + (LN B).
Para quem já estiver familiarizado com logaritmos vulgares, LN é o logaritmo natural. O logaritmo usual é um múltiplo deste, chamado log X = (LN X)/(LN 10).
Historicamente, o logaritmo foi usado para inverter os totais de multiplicações em adições. Com o advento dos computadores, esta técnica passou a ser irrelevante, mas EXP e LN continuam a ter a sua importancia por outras razões. Por exemplo, EXP simboliza-se por uma curva - chamada catenária.

```
10 FOR J = 0 TO 60
20 PLOT J,(EXP(.1*(J-30))+EXP(.1 (30-J)))*
   *1,5
30 NEXT J
```

Para uma ilustração final, pense naquilo a que os técnicos de demografia chamam "crescimento potencial". A idéia é que a população do mundo aumenta em cada ano, tal como um juro composto, de modo que ao fim de N anos, uma população original P passa a ser

P*((1*R)**N) em que R é a taxa de crescimento.
Em 1220, Leonardo Fibonacci criou a sua famosa sequencia de números:
0112358 13 21 34 55 89 144...
Cada número é a soma dos dois anteriores. Este foi o modelo de crescimento de uma população de coelhos. Os números crescem sempre potencialmente.
O programa que vamos agora apresentar aplica o kª número de Fibronacci, F(K) e também a expressão (LN F(K))/K por razões que explicitaremos:

```
10 LET B= 1
20 LET N= 0
30 LET K= 0
40 LET C= B
50 LET B= B+N
60 LET N= C
70 LET K= K+1
80 PRINT B,(LN B)/K
90 IF K >20 THEN SCROLL
100 GOTO 40
```

A segunda coluna de números é estabelecida para um valor de R, aproximadamente 0,4794403. Portanto a Kª geração de coelhos tem F(K) membros, em que

(LN F(K))/K = R.

Então:

LN F(K) = K*R

Portanto:

F(K) = EXP (K*R) = (EXP R)**K

Teoricamente, o valor para EXP (R) é o número (1+ SQR 5)/2.
Para uma rápida confirmação, ordenamos

PRINT LN ((1+SQR 5)/2

e obtemos 0,48121183, que está bastante próximo.

SE QUER OBTER O LOGARITMO DE BASE 10 :

```
9000  LET X = LN X / LN 10
8000 INPUT X
9100 PRINT X
9200 GO TO 8000
```

```
4687 REM
4690 PRINT "S/ IT INCLUIDO -25"
4695 PRINT M$
4700 INPUT A$
4703 IF A$="0"THEN GOTO 5
4705 LET C=VAL A$
4715 LET B=VAL "(INT ((1.4375*C)+0.6)/10)*10"
4720 LET U$=STR$ B
4725 LET U=LEN U$
4730 LET V=LEN A$
4735 PRINT TAB 10-V;A$;TAB 25-U;B;".0"
4737 GOTO 4700
4740 CLS
4745 PRINT "S/ IT INCLUIDO -30"
4750 PRINT M$
4755 INPUT A$
4760 IF A$="0"THEN GOTO 5
4765 LET C=VAL A$
4770 LET B=VAL "(INT ((1.495*C)+0.6)/10)*10"
4775 LET U$=STR$ B
4780 LET U=LEN U$
4785 LET V=LEN A$
4790 PRINT TAB 10-V;A$;TAB 25-U;B;".0"
4800 GOTO 4755
4802 GOSUB 8000
8040 RETURN
```

Este pequeno programa chama-se PADRÃO UNIVERSAL e é capaz de produzir mais de dois milhões de desenhos distintos no seu T V ou na sua impressora.
Cada matriz tem um numero e o comprimento da string é impresso na ultima linha, de modo que possa reproduzir o mesmo desenho em qualquer altura. Para gerar uma nova matriz, responda "S". Tome nota dos numeros que aparecem na ultima linha. Se quiser a mesma matriz, responda "N" e dê entrada dos numeros anotados.

```
  10 DIM A(33)
  15 SLOW
  20 PRINT "MATRIZ ALEATORIA ? (S/N)"
  30 INPUT A$
  40 CLS
  50 IF NOT A$="S"THEN GOTO 200
  60 LET L=INT (RND*33)+1
  70 LET X=INT (RND*65535)+1
  80 RAND X
  85 FAST
  90 FOR J=1TO L
 100 LET A(J)=INT (RND*10)+1
 110 NEXT J
 120 FOR J=1TO 660/L
 130 FOR K=1TO L
 140 PRINT CHR$ A(K);
 150 NEXT K
 160 NEXT J
 170 PRINT X;"(";L;")"
 180 STOP
 200 PRINT "NUMERO DA MATRIZ ?"
 210 INPUT X
 220 CLS
 230 PRINT "COMPRIMENTO DA STRING ?"
 240 INPUT L
 250 CLS
 260 GOTO 80
 900 REM
 910 REM
```

PROGRAMA

" MORSE"

ESTE PROGRAMA, QUE NOS FOI CEDIDO PELO Dr.LUIS LEITÃO E QUE É PROVENIENTE DO Sinclair Amateur Radio User Group permite escrever uma mensagem no seu T V e se mantiver o seu gravador ligado com o cabo " MICRO", e na posição "RECORD", terá a sua mensagem gravada na fita e em morse. Poderá depois reproduzir a mensagem e tentar interpretar o código, como treino.

```
   1 REM PEEK TO TAN
   2 REM ...................................
..........
  10 REM "MORSE"
  20 LET B=8
  25 LET S=24
  30 SLOW
  50 IF INKEY$<>""THEN GOTO 50
  60 IF INKEY$=""THEN GOTO 60
  70 LET A$=INKEY$
  80 PRINT A$;
  85 IF A$="."THEN GOTO 30
  90 FAST
 100 LET C=CODE A$-27
 120 LET E=PEEK 16522+C
 250 LET E2=INT (E/2)
 300 LET E3=E-(E2*2)
 350 LET Y=B+(B*2*(E3=1))
 450 FOR X=1TO Y
 500 RAND USR 16514
 550 NEXT X
 600 FOR X=1TO (120-240*(E<2))/S
 650 NEXT X
 700 LET E=E2
 750 IF E<=1THEN GOTO 30
 800 GOTO 250
 850 STOP
1000 FOR I=16523TO 16559
1010 INPUT X
1030 POKE I,X
1040 NEXT I
1090 STOP
```

INSTRUÇÕES : Depois de introduzir o programa, deve fazer GO TO 1000 e introduzir, um a um, os seguintes códigos :

63 62 6o 56 48 32 33 35 39 47 6 17 21 9 2 2o 11
16 4 3o 13 18 7 6 15 22 27 lo 8 5 12 24 14 25 29 19 42

Pode agora fazer RUN e escrever a sua mensagem quando o ecran aparecer branco.

Pensamos poder apresentar no próximo mês, um pequeno circuito que lhe permita escutar simultâneamente com o escrever da mensagem, o código " MORSE".

BATALHA NAVAL

```
  REM BATALHA NAVAL Trad. LOG Tecnologia Industrial Lda.
            12/10/82
   1 REM "BATALHA"
  10 PRINT AT 9,0;"--------------------------------"
  20 PRINT AT 10,10;"BATALHA NAVAL"
  30 PRINT AT 11,0;"--------------------------------"
  40 PRINT AT 19,25;"<log>"
  50 PAUSE 200
  60 CLS
 400 PRINT AT 19,25;"<log>"
1000 LET S=0
1010 LET K$="                              "
1020 DIM B$(10,2)
1030 LET M=0
1040 LET N=0
1050 LET J=1
1060 GOSUB 1090
1070 GOTO 1230
1090 FAST
1110 FOR X=0TO 16STEP 16
1120 FOR I=4TO 13
1130 FOR Z=2TO 11
1140 PRINT AT I,Z+X;"."
1150 PRINT AT I,Z+X;"."
1160 NEXT Z
1170 PRINT AT I,0;TAB X;I-4
1180 NEXT I
1190 PRINT AT 3,2+X;"ABCDEFGHIJ"
1200 NEXT X
1210 SLOW
1220 RETURN
1230 PRINT AT 21,0;"  entrada das coordenadas  "
1240 PAUSE 250
1245 PRINT AT 21,0;K$
1250 FOR A=1TO 4
1255 PRINT AT 20,0;K$
1260 PRINT "->POSICAO DO CRUZADOR N. ";A
1265 INPUT B$(A)
1267 IF CODE B$(A)<38OR CODE B$(A)>47THEN GOTO 1265
1270 PRINT AT CODE (B$(A,2))-28+4,CODE (B$(A,1))-38+2;"C"
1280 NEXT A
1290 FOR A=5TO 8
1300 PRINT AT 21,0;"->POSICAO DA FRAGATA  N. ";A-4
1310 INPUT B$(A)
1315 IF CODE B$(A)<38OR CODE B$(A)>47THEN GOTO 1310
1320 PRINT AT CODE (B$(A,2))-28+4,CODE (B$(A,1))-30+2;"F"
1330 NEXT A
1340 FOR A=9TO 10
1350 PRINT AT 21,0;"->POSICAO DO SUBMARINO N. ";A-8
1360 INPUT B$(A)
1365 IF CODE B$(A)<38OR CODE B$(A)>47THEN GOTO 1360
1370 PRINT AT CODE (B$(A,2))-28+4,CODE (B$(A,1))-38+2;"S"
1380 NEXT A
1400 DIM A$(10,2)
1410 FOR A=1TO 4
1420 LET A$(A,1)=CHR$ (INT (RND*10)+38)
1430 LET A$(A,2)=CHR$ (INT (RND*10)+28)
1440 NEXT A
1450 CLS
1460 FOR A=5TO 8
1470 LET A$(A,1)=CHR$ (INT (RND*10)+38)
1480 LET A$(A,2)=CHR$ (INT (RND*10)+28)
```

```
1490 NEXT A
1500 FOR A=9TO 10
1510 LET A$(A,1)=CHR$ (INT (RND*10)+38)
1520 LET A$(A,2)=CHR$ (INT (RND*10)+28)
1530 NEXT A
1540 GOSUB 1090
1550 FOR A=1TO 4
1560 PRINT AT CODE (B$(A,2))-28+4,CODE (B$(A,1))-38+2;"C"
1570 NEXT A
1580 PRINT AT 0,0;"OS SEUS BARCOS"
1590 FOR A=5TO 8
1600 PRINT AT CODE (B$(A,2))-28+4,CODE (B$(A,1))-38+2;"F"
1610 NEXT A
1620 FOR A=9TO 10
1630 PRINT AT CODE (B$(A,2))-28+4,CODE (B$(A,1))-38+2;"S"
1640 NEXT A
1650 PRINT AT 21,0;K$;AT 21,0;"disparo"
1660 PRINT AT 19,0;"NOSSA ESQUADRA= ";S;AT 20,0;"ESQUADRA INIMIGA= ";M
1670 FOR B=1TO 3
1680 PRINT AT 15,0;K$
1690 LET G=-150
1700 INPUT C$
1703 LET N=N+1
1705 FOR A=1TO 10
1707 IF C$=A$(A)THEN PRINT AT 1,15;"CERTO:";C$
1713 IF C$<>A$(A)THEN PRINT AT 1,15;"ERROU:";C$
1720 IF C$=A$(A)THEN GOTO 1810
1740 NEXT A
1750 PRINT AT CODE (C$(2))-28+4,CODE (C$(1))-38+18;"+"
1760 LET K=RND**RND
1790 NEXT B
1800 GOTO 1960
1810 IF G=145THEN PRINT AT 15,0;"foi atingido"
1820 IF G=-150THEN PRINT AT 15,0;"fui atingido"
1830 IF A<=4THEN PRINT "cruzador"
1850 IF A>=5AND A<=8THEN PRINT "fragata"
1860 IF A>=9AND A<=10THEN PRINT "submarino"
1870 IF G=145THEN LET M=M+1
1880 IF G=-150THEN PRINT AT CODE (C$(2))-28+4,CODE (C$(1))-38+18;"h"
1890 IF G=145THEN PRINT AT CODE (D$(B,2))-28+4,CODE (D$(B,1))-38+2;"h"
1900 IF M=10THEN PRINT AT 20,20;"GANHEI"
1904 IF M=10THEN PAUSE 250
1908 IF M=10THEN GOTO 0
1910 IF S=10THEN PRINT AT 19,19;"GANHOU"
1914 IF S=10THEN PAUSE 250
1918 IF S=10THEN GOTO 0
1920 IF G=-150THEN LET A$(A)=""
1930 IF G=145THEN LET B$(A)=""
1940 LET K=RND**RND**RND
1950 GOTO 1930+G-5
1960 PRINT AT 21,0;K$;AT 21,0;" zx81 "
1970 DIM D$(3,2)
1980 LET G=145
1990 FOR B=1TO 3
2000 LET D$(B,1)=CHR$ (INT (RND*10)+38)
2010 LET D$(B,2)=CHR$ (INT (RND*10)+28)
2020 FOR A=1TO 10
2025 IF D$(B)=B$(A)THEN PRINT AT 1,15;"CERTO:";D$(B)
2030 IF D$(B)<>B$(A)THEN PRINT AT 1,15;"ERROU:";D$(B)
2035 IF D$(B)=B$(A)THEN GOTO 1810
2040 NEXT A
2055 PRINT AT CODE (D$(B,2))-28+4,CODE (D$(B,1))-38+2;"+"
2070 NEXT B
2080 PRINT AT 17,0;"            "
2090 GOTO 1650
2100 SAVE "BATALHA"
```

COBOL (Common Business Oriented Language) - Linguagem de Alto Nível, usada em processamento de problemas principalmente do ramo comercial

COMPILER - Programa cuja função é ler outro programa escrito numa linguagem de alto nível, tal como o COBOL ou FORTRAN, convertendo-o num código máquina a que o computador possa obedecer.

CP/M (Control Program/Microprocessor) - Sistema operativo com discos para microcomputadores, usando os processadores 8080 e Z80.

CPS (ou CHPS) - Caracteres por segundo.

CUTS (Computer Users' Tape System) - modelo para gravação de dados em cassete.

DAISY WHEEL - Componente de uma impressora sequencial, cujos caracteres estão na periferia de um disco plástico.

DATA BASE - Sistema para organizar elementos de informação numa série de códigos máquina, de tal modo que um programa possa imediatamente seleccionar qualquer abstracção particular ou combinação de informação que venha a ser chamada. P. ex., um cliente de uma base de dados pode incluir detalhes completos de todos os clientes (conforme requerido por departamentos de serviços e distribuição ou vendas e marketing) e também de toda a entrega e pedidos de serviço, assim como cada artigo facturado para esses clientes no espaço de um ano ou mais. Um programa posterior pode permitir o acesso a bases de dados sobre questões do tipo " identificação dos clientes que compram mais de 1 000 contos de um artigo 'A' em menos de 5 entregas, e que recebem menos de 2 pedidos de serviço no ano".

DEBUG - Para corrigir erros num programa.

DISC (DISK) - Dispositivo de armazenamento magnético que permite um rápido acessó aleatório a qualquer selecção de um grande volume de dados. Um disco rígido de grandes dimensões contém aproximadamente 5 "megabytes" ou mais, um "floppy disc" geralmente contém entre 80 a 250 "kilobytes", mas em qualquer dos casos a capacidade vai aumentando a todo o tempo.

DISKETTE - "Floppy disc", especialmente o de tamanho mais pequeno 5 1/4"

DOS (Disc Operating Sistem) - Sistema operativo por computador, cujo suporte é um disco magnético em vez da memória ROM. Um processo inicial copiará o sistema operativo para a memória sempre que o computador seja ligado. É também um sistema operativo que controla os próprios discos e que pode completar, em vez de substituir, o sistema da operação inicial.

DUPLEX - Tipo de transmissão de dados no qual cada estação pode enviar e receber simultaneamente.

DYNAMIC (Memória) - Memória de Acesso Aleatório (RAM) que requer frequentes sinais de realimentação, mas que, normalmente, gasta menos energia que a memória estática.

EAROM (Electrically Alterable Read Only Memory) - Levando geralmente 10 mseg. para apagar, e 1 mseg. para escrever, este armazenamento não-volátil deveria ser melhor classificado como "Read Mostly Memory" (memória quase exclusivamente de leitura), dado que a sua capacidade atinge um limite de aproximadamente 100 000 ciclos.

EDIT - Alteração do texto no programa ou séries de dados. Quase sempre, alguns sistemas fazem edições mais facilmente que outros.

EPROM (Erasable Programmable Read Only Memory) - Demora normalmente um minuto a escrever e 10 ou mais minutos a apagar, através de raios ultra-violetas.

FIRMWARE - Instruções ou dados permanentemente armazenados em ROM.

FLOPPY (disco) - Memória de massa que inclui um disco flexível com uma superfície magnética, onde os dados são gravados, e aos quais se tem rápido acesso movendo a cabeça de leitura/escrita. O disco, que pode ter 8" ou 5 1/4" de diametro, roda no interior de uma embalagem protectora.

FORTRAN (Formula Translation) - Linguagem de programação de alto nível que, apesar de não ser recente, é ainda muito usada principalmente no mundo científico.

HARD COPY - Cópia em papel tirada por uma impressora.

HARDWARE - Elementos físicos de um computador (em oposição a "software")

HIGH-LEVEL LANGUAGE - Linguagem de programação semelhante à linguagem natural e com instruções poderosas que produzem várias instruções em linguagem máquina. Alguns exemplos: BASIC, COBOL e FORTRAN.

INTELLIGENT TERMINAL - Um dispositivo de entrada e saída, que inclui os seus próprios circuitos lógicos e memória, de modo que, p. ex., a disposição dos dados pode ser validada ou alterada antes da transmissão para o computador central.

Lembrámos já no boletim nº zero, e voltamos agora a insistir, que o Clube Z-80 só pode efectivamente prosseguir os seus objectivos - nomeadamente a publicação e distribuição dos boletins aos seus membros - se possuir os necessários meios económicos.
Após a divulgação no semanário "O Jornal" (15.10.82) da existencia do Clube Z-80 , vários utilizadores de ZX81 de todo o país nos tem contactado, felicitando-nos por esta iniciativa e manifestando interesse em associar--se. Não obstante, as pessoas que neste momento estão inscritas não são ainda suficientes para cobrir as despesas que uma tarefa deste tipo implica.
O futuro do Clube depende de todos nós. Por isso, se realmente está interessado em ser um dos seus membros, solicitamos-lhe que faça a sua inscrição e pagamento - pela modalidade que mais lhe convier (anual, semestral ou trimestral).

Por razões que decerto todos compreenderão, o boletim nº 2, a ser publicado em Novembro, apenas será distribuído às pessoas que tiverem confirmado a sua inscrição.

Anexamos um cupão de inscrição, ficando a aguardar a sua resposta.

* INSCRIÇÃO no CLUBE Z - 80 — Z-80

NOME...

IDADE..... ENDEREÇO.....................................

....

COMPUTADOR TIPO...

PROFISSÃO...

SUBSCRIÇÃO	ANUAL	SEMESTRAL	TRIMESTRAL
	1 500$00	750$00	375$00
	☐	☐	☐

CHEQUE Nº.................S/..............................

DATA.................. *